Mining Users Skills Development From Interaction Traces : an exploratory study

Amaury Belin

University of Lyon, CNRS LIRIS, UMR5205, F-69622 amaury.belin@liris.cnrs.fr

ABSTRACT

In this paper, we report an exploratory work related to the detection of user skills development from large interactions traces. We present a statistical approach to tackle the development of user's low level skills by identifying patterns of action and studying their progressive automations (i.e. as a user develops skills, they execute common subsequent actions faster) as well as their evolutions (i.e. user elaborating innovative ways to achieve a common task).

Mots Clés

Interaction trace; user skills; data mining; frequent episode mining.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

CONTEXT

Our work is based on low-level interaction traces of a student using the video annotation *Lignes de Temps* (LDT) to carry out a research project focusing on 15 films. It was the first time she was using the software and her work spanned over 3 months. In addition to interaction traces, on-field observations and interviews allowed us to assess different phenomenon related to skills acquisition and progressive mastering of LDT by the user.

DATA: INTERACTION TRACE

An interaction trace is composed of *obsels* representing events occurring during the interaction between a user and a particular software [3]. Each *obsel* has timing attributes (i.e. the date and duration of the represented event), as well as attributes related to its type, different *obsel types* being defined to represent the variety of events that could occur in the interaction (e.g. "playVideo", "createAnnotation", etc.). In our experiment, the complete trace contains 50,000 obsels, representing 50 hours of work spanned over the first 15 days the user spent with the software. These obsels Grigory Antipov, Julien Blanchard, Fabrice Guillet and Yannick Prié University of Nantes, CNRS LINA UMR CNRS 6241, antigregory@gmail.com, {julien.blanchard, fabrice.guillet, yannick.prie }@univ-nantes.fr

belong to 89 obsel types. We only considered the 34 types representing user actions, the other types being dedicated to the description of system processing in reaction to user actions (e.g. "videoStartPlaying").

PATTERN ELICITATION

At first, we analysed interaction traces with simple visualization tools. Guided by the other materials we collected (interviews, screen recordings, etc.), we managed to detect some phenomenon related to skills acquisition in the trace [1]. In particular, we identified some schemes of action by the user, i.e. systematic ways to achieve a goal by reproducing a similar construction in a set of articulated actions. Some of these schemes were easily noticeable in the visualization of the interaction traces. In Figure 1, we can see that the user is reproducing the same scheme: in order to characterize a part of the movie she is working on, she selects the corresponding annotation, plays and then pauses the movie to watch the segment corresponding to the annotation, and finally enters different tags in a text field. We see that the user reproduces the same pattern successively, for each part of the movie she has to deal with.



characterized by an expert using a color codification The apparition of such pattern is a clear hint of user appropriation of the software. However identifying every pattern is a tedious work that could be facilitated by a statistical approach. With this perspective in sight, we have chosen to use a *frequent episode mining* algorithm [4] to analyse the interaction traces. Frequent episode mining is an essential task in data mining, which allows for detecting frequent patterns (called episodes) in a

single sequence of events. This first attempt provided us

with numerous patterns that we analysed given the two approaches presented below.

EVOLUTION OF PATTERN DURATION

Analysing the occurrences of the annotation-tagging pattern (see last section) reveals that the user is progressively getting faster at chaining-up the different actions, and seems to execute the pattern in a more systematic and automatized way (see pattern characterization in Figure 1). Therefore, we studied the evolution of the duration of the occurrences of the patterns we detected, with the idea that a monotonous decreasing of an occurrence's duration of a particular pattern might point out skill acquisition regarding this pattern. We refined this approach by discarding the part of the duration that could be dependant on the content the user is dealing with. For instance, in the pattern consisting of tagging an annotation, execution time is dependant of the length of the movie segment being annotated (as the pattern involves the user to play this segment) as well as the complexity of the tags being edited. Therefore, in our measure of pattern occurrence duration, we discarded the part of the pattern that can be dependant of the content being addressed.



Figure 2. Evolution of the duration of a particular pattern, with regression coefficient line (in red).

For each identified patterns, we measured the *content-independent* duration of each occurrences, and calculated the regression coefficient of the distribution of these durations (see Figure 2). We analysed 60 patterns having regression models presenting potentially interesting coefficients. We did not found straightforward evident results, but indirectly this first attempt guided us to investigate part of the practice where skills acquisition might occur.

PATTERN VARIATIONS

A pattern evolution is an interesting phenomenon as it can be a cue that the user has learned a more elaborated way to accomplish a common task. For instance, we noticed an evolution of the *annotation-tagging pattern* related above: as the user experimented crashes of the software and lost some work, she was systematically saving her work after the edition of an annotation tag, leading to a noticeable evolution of the pattern. We did therefore look for a statistical mean that would help to address such pattern evolutions. Our approach was to study the distribution of the occurrences of similar patterns. Here we made two assumptions: if two patterns are close in their composition, they may be used for the same task; and if two patterns occurrences are grouped in different periods of the activity, it could be the expression that one more elaborated version of a pattern has replaced a former version.

In order to measure similarity between patterns, we used the Hamming distance [2]. We refined such distance by valuating substitution between actions composing the pattern based on the similarity of these actions. We then plotted occurrences of similar patterns (see Figure 3) and tried to interpret if one pattern could have replaced another in the user's practice, and if we can speak of a pattern evolution. However we didn't find clearly interesting results yet with this approach.



Figure 3. Distribution of the occurrences of similar patterns over several days.

CONCLUSION

Using interaction traces as a material to study processes occurring over long period of time (like skill development in our case) is challenging. The volume of data is large and rich, and there is a real difficulty to synthetize automatically collected low-level interaction events to higher level information about user activity [3].

We presented an overview of our exploratory attempt to address such challenge, and we introduced some of the ideas we came up with. Our approach should now be refined with some heuristics to pre-filter the amount of detected patterns and focus the analysis on more fertile part of the interaction trace. It also also raises numerous questions on how to interactively visualize and mine interaction traces to keep track of user skills development and, more generally, other slow processes underlying human-computer interaction.

BIBLIOGRAPHIE

- Belin, A. and Prié, Y. DIAM: towards a model for describing appropriation processes through the evolution of digital artifacts. *Proceedings of the Designing Interactive Systems Conference*, ACM (2012), 645–654.
- 2. Hamming, R.W. *Coding and information theory*. Prentice-Hall, Inc., 1986.
- 3. Hilbert, D.M. and Redmiles, D.F. Extracting usability information from user interface events. *ACM Computing Surveys (CSUR)* 32, 4 (2000), 384–421.
- 4. Mannila, H., Toivonen, H., and Verkamo, A.I. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery 1*, 3 (1997), 259–289.